

**期末项目设计报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 题 目 | 基于软件项目的托管平台的实验管理平台的分析与设计 | | |
| 课程 | 软件系统分析与设计技术 | | |
| 学 院 | 信息科学与工程学院 | | |
| 专 业 | 软件工程 | 年级 | 2016级 |
| 学生姓名 | 吴伟辉 | 学号 | 201610414325 |
| 指导教师 | 赵卫东 | 职称 | 副教授 |

|  |  |  |  |
| --- | --- | --- | --- |
| **评分项** | **评分标准** | **满分** | **得分** |
| 文档整体 | 文档内容详实、规范，美观大方 | 20 |  |
| 用例图及规约 | 用例图完整，准确，能够完全体现需求 | 10 |  |
| 顺序图与活动图 | 顺序图能够完全描述用例的设计思路和业务流程，活动图能够描述部分算法的流程。 | 10 |  |
| 类图 | 类图能够完整，准确反映业务的需数据的组织结构 | 10 |  |
| 数据库 | 数据库表的设计来自于类图，合理 | 10 |  |
| 界面设计 | 界面设计美观，清晰，合理，能够完全反映用例图的设计思路 | 10 |  |
| API接口设计 | 为每个界面设计合理的API接口，以便进行前后端分离开发 | 15 |  |
| 内容一致性 | 用例图，类图、数据库、界面相互印证，相互依赖，环环相扣 | 15 |  |
| **得分合计** | | |  |

2019年5月1日

目录

[1. 概述](#_Toc32723_WPSOffice_Level1) [3](#_Toc32723_WPSOffice_Level1)

[2. 系统总体结构](#_Toc18364_WPSOffice_Level1) [4](#_Toc18364_WPSOffice_Level1)

[3. 用例图设计](#_Toc20225_WPSOffice_Level1) [4](#_Toc20225_WPSOffice_Level1)

[4. 类图设计](#_Toc21280_WPSOffice_Level1) [5](#_Toc21280_WPSOffice_Level1)

[5. 数据库设计](#_Toc5164_WPSOffice_Level1) [5](#_Toc5164_WPSOffice_Level1)

[6. 用例及界面详细设计](#_Toc22335_WPSOffice_Level1) [8](#_Toc22335_WPSOffice_Level1)

[7. 接口](#_Toc28693_WPSOffice_Level1) [17](#_Toc28693_WPSOffice_Level1)

## 基于软件项目的托管平台的实验管理平台的分析与设计

### 1. 概述

基于软件项目的托管平台的实验管理平台的作用是在线管理实验成绩的Web应用系统。学生和老师的实验内容均存放在Github,Gitlab,Gitee等页面上。

学生的功能主要有：一是设置自己的Github,Gitlab,Gitee等用户名，二是查询自己的实验成绩。学生的Github,Gitlab,Gitee等用户名是公开的，但成绩不公开。

学生可以在实验平台中绑定帐号，如果有多个帐号，可以选择提交作业的帐号。

老师的功能主要有：一是批改每个学生的成绩，二是查看每个学生的成绩。

老师和学生都能通过本系统的链接方便地跳转到学生的每个Github,Gitlab,Gitee实验目录，以便批改实验或者查看实验情况。

老师可以开多门课程，学生可以选择多门课程，学生选课时需要老师开了次课程。

老师在评分的时候可以给每一个小题设置分数，最后分数由系统自加,每个小题也可以由教室设置分值与数量。

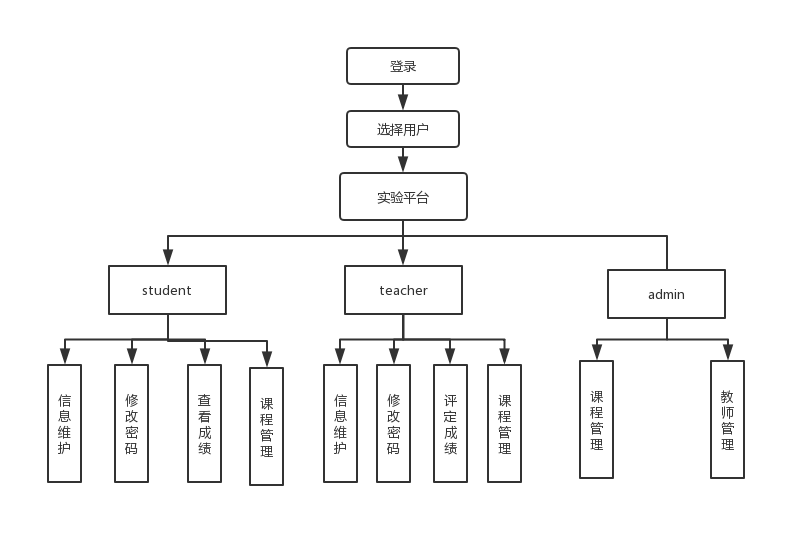
实验成绩按数字分数计算，每项实验的满分为100分，最低为0分。

学期结束后该课程不会结束，下学期可以让其他学生选择该课程。

老师可以根据学科要求设置实验次数和实验分值。

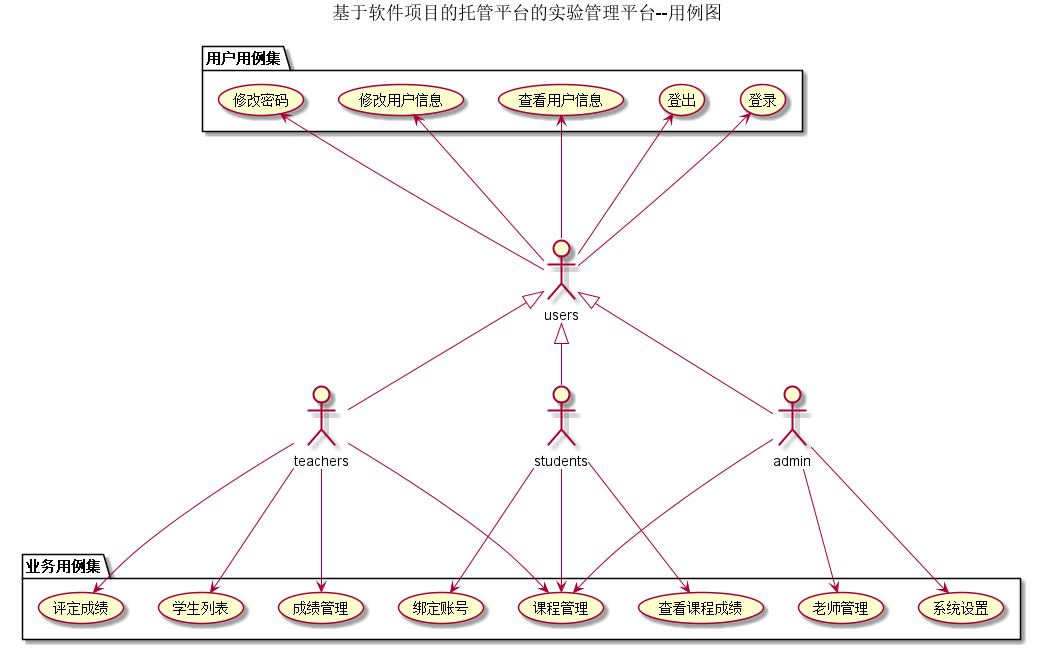
系统自动计算每个学生的所有实验的平均分。

### 2. 系统总体结构

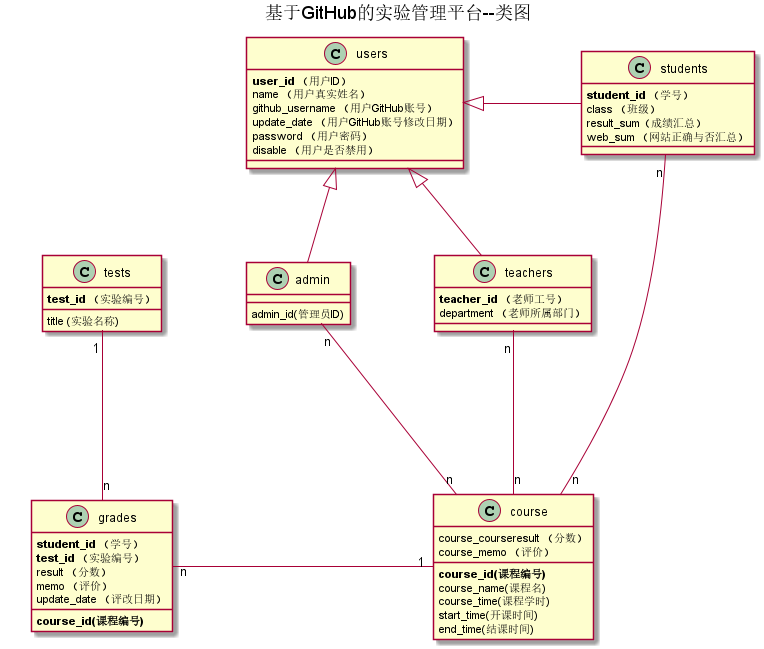


界面设计参见：https://wwhkiller.github.io/is\_analysis\_pages/test6\_Ui/index.html

### 3. 用例图设计



### 4. 类图设计



### 5. 数据库设计

USERS表（用户表）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **主键， 外键** | **可以为空** | **默认值** | **约束** | **说明** |
| USER\_ID | NUMBER(8,0) | 主键 | 否 |  |  | 用户ID |
| NAME | VARCHAR2(50 BYTE) |  | 否 |  |  | 用户真实姓名 |
| GITHUB\_USERNAME | VARCHAR2(50 BYTE) |  | 是 | 空 |  | GitHUB用户名 |
| UPDATE\_DATE | DATE |  | 是 | 空 |  | GitHUB用户名修改日期 |
| PASSWORD | VARCHAR2(512 BYTE) |  | 是 | 空 |  | 加密存储密码，为空表示密码就是学号 |
| DISABLE | VARCHAR2(20 BYTE) |  | 否 |  |  | 是否禁用,值为是表示禁用,其他表示正常. |

TEACHERS表（老师表）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **主键， 外键** | **可以为空** | **默认值** | **约束** | **说明** |
| TEACHER\_ID | VARCHAR2(50 BYTE) | 主键 | 否 |  |  | 老师的编号 |
| USER\_ID | NUMBER(8,0) | 外键 | 是 |  |  | 老师的用户ID，USERS 表的外键 |
| DEPARTMENT | VARCHAR2(400 BYTE) |  | 否 |  |  | 老师属于的部门 |

STUDENTS表（学生表）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **主键， 外键** | **可以为空** | **默认值** | **约束** | **说明** |
| STUDENT\_ID | VARCHAR2(50 BYTE) | 主键 | 否 |  |  | 学生的学号 |
| USER\_ID | NUMBER(8,0) | 外键 | 是 |  | 空 | 学生的用户ID，USERS表的外键，为空表示还没有 建立用户 |
| MA JOR | VARCHAR2(20 BYTE) |  | 否 |  |  | 学生的专业 |
| CLASSNUM | VARCHAR2(20 BYTE) |  | 否 |  |  | 学生的班级号 |
| RESULT\_SUM | VARCHAR2(400 BYTE) | 外键 | 是 | 空 |  | 成绩汇总（来自GRADES表），以逗号分开，第一 个成绩是平均成绩,后面是每次实验的成绩，N表示未批改，平均分只计算已批改的。比  如：“81.25,70,80,85,90,N”表示一共批改了4次， 第5次未批改，4次的成绩分别是81.25,70,80,85,90,N，4次的平均分是81.25 |
| WEB\_SUM | VARCHAR2(400 BYTE) |  | 是 | 空 |  | GitHub网址是否正确，用逗号分开，Y代表正确， N代表不正确。第1位代表总的GitHUB地址是否正确，第2位表示第1次实验的地址，第3位表示第2 位实验地址，依此类推。比如：“Y,Y,Y,Y,Y,N”表示第 5次实验地址不正确，其他地址正确 |

GRADES表（学生实验成绩表）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **主键，外键** | **可以为空** | **默认值** | **约束** | **说明** |
| STUDENT\_ID | VARCHAR2(50 BYTE) | 联合主键  1，外键 | 否 |  |  | 学生的学号，  STUDENTS表外键 |
| TEST\_ID | NUMBER(6,0) | 联合主键  2，外键 | 否 |  |  | 实验编号，TESTS表的外键 |
| RESULT | NUMBER | 主键 | 是 | 空 | 取值  0-  100 | 分数，这个值为空表示没有批改 |
| MEMO | VARCHAR2(400 BYTE) |  | 是 | 空 |  | 老师对实验的评语 |
| UPDATE\_DATE | DATE |  | 是 | 空 |  | 老师批改实验的日期， 为空表示未批改 |

TESTS表（实验项目表）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **主键，外键** | **可以为空** | **默认值** | **约束** | **说明** |
| TEST\_ID | NUMBER(6,0) | 主键 | 否 |  |  | 实验编号 |
| TITLE | VARCHAR2(100 BYTE) |  | 否 |  |  | 实验名称 |

COURSEB表(课程表)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **主键，外键** | **可以为空** | **默认值** | **约束** | **说明** |
| COURSEB\_ID | NUMBER(6,0) | 主键 | 否 |  |  | 课程编号 |
| COURSEB\_NAME | VARCHAR2(100 BYTE) |  | 否 |  |  | 课程名称 |
| COURSEB\_TIME | INT |  | 否 |  |  | 学时 |
| START\_TIME | DATE |  | 否 |  |  | 开课时间 |
| END\_TIME | DATE |  | 否 |  |  | 结课时间 |

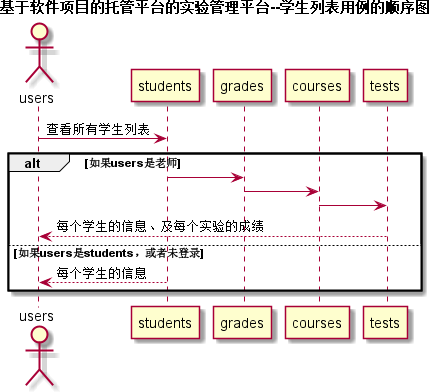
### 6. 用例及界面详细设计

#### “学生列表”用例

-用例规约

|  |  |
| --- | --- |
| 用例名称 | 学生列表 |
| 功能 | 以表形式的显示出所有学生的信息 |
| 参与者 | 游客，老师 |
| 前置条件 | 老师需要先登录 |
| 后置条件 |  |
| 主事件流 |  |
| 备选事件流 |  |

-业务流程（顺序图）



-界面设计

界面参照: <https://wwhkiller.github.io/is_analysis_pages/test6_Ui/index.html> <https://wwhkiller.github.io/is_analysis_pages/test6_Ui/teacher.html>

API接口调用

接口1：getStudents

算法描述

WEB\_SUM解析为列表

WEB\_SUM是接口getStudents的返回值中的一部分，形如："WEB\_SUM": "Y,Y,Y,Y,Y,N"。需要在前端进行解析。

作用是判断GitHub网址是否正确，用逗号分开，Y代表正确，N代表不正确。

第1位代表总的GitHUB地址是否正确，第2位表示第1次实验的地址，第3位表示第2位实验地址，依此类 推。

比如：“Y,Y,Y,Y,Y,N”表示第5次实验地址不正确，其他地址正确。算法是使用使用字符串分离方法将其分开。

RESULT\_SUM解析为列表

RESULT\_SUM是接口getStudents的返回值中的一部分，形如："RESULT\_SUM": "81.25,70,80,85,90,N"。需要在前端进行解析。

成绩汇总（来自GRADES表），以逗号分开，第一个成绩是平均成绩,后面是每次实验的成绩，N表示未批改，平均分只计算已批改的。

比如：“81.25,70,80,85,90,N”表示一共批改了4次，第5次未批改，4次的成绩分别是

81.25,70,80,85,90,N，4次的平均分是81.25

算法是使用使用字符串分离方法将其分开。

-参照表

STUDENTS

GRADES

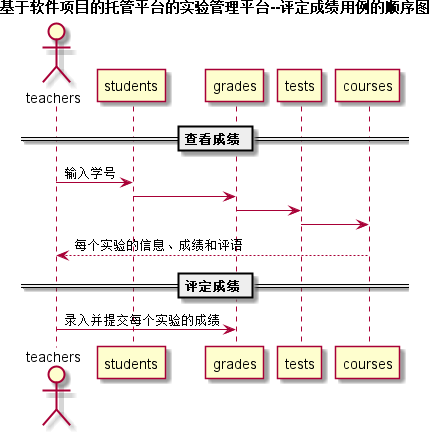
TESTS

“评定成绩”用例

-用例规约

|  |  |
| --- | --- |
| 用例名称 | 评定成绩 |
| 功能 | 老师评定一个学生的实验成绩 |
| 参与者 | 老师 |
| 前置条件 | 查看成绩：评定成绩之前，老师需要先登录，并显示出一个学生的所有实验成绩和评语信息 |
| 后置条件 | 评定成绩提交之后，系统自动设置成绩更新日期为当前日期，自动计算平均成绩 |
| 主事件流 | 查看已有的成绩  输入一个或者多个实验的成绩和评语  提交  系统存储实验成绩和评语  系统自动计算平均成绩 |
| 备选事件流 | 成绩必须是0至100之间，可以为空，但不能超界  提示用户重新输入成绩分数 |

-业务流程（顺序图） 源码



- 界面设计

界面参照: <https://wwhkiller.github.io/is_analysis_pages/test6_Ui/teacher.html> API接口调用

接口1：getNextPrevStudent

用于取得上一个或者下一个学生的学号接口2：getOneStudentResults

用于显示一个学生的所有实验成绩和评语

接口3：setOneStudentResults

用于设置一个学生的部分实验成绩和评语

-算法描述

无

-参照表

STUDENTS

GRADES

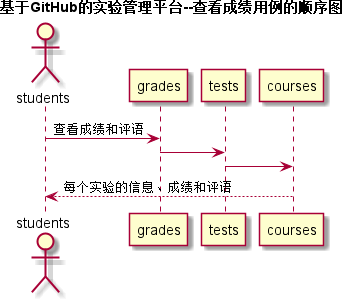
TESTS

#### “查看成绩”用例

-用例规约

|  |  |
| --- | --- |
| 用例名称 | 查看成绩 |
| 功能 | 学生查看自己的每个实验的实验成绩及实验评价 |
| 参与者 | 学生 |
| 前置条件 | 学生需要先登录 |
| 后置条件 |  |
| 主事件流 |  |
| 备选事件流 |  |

-业务流程（顺序图） 源码



-界面设计

界面参照: <https://wwhkiller.github.io/is_analysis_pages/test6_Ui/student.html> API接口调用

接口1：getOneStudentResults

-算法描述

无

-参照表

STUDENTS

GRADES

TESTS

#### “修改密码”用例

-用例规约

|  |  |
| --- | --- |
| 用例名称 | 修改密码 |
| 功能 | 修改用户的密码 |
| 参与者 | 学生，老师 |
| 前置条件 | 必须先登录 |
| 后置条件 | 修改密码以后必须强制登出，再跳转到登录页面 |
| 主事件流 | 用户填写新密码（两次输入）  用户提交修改信息  系统存储修改后密码 |
| 备选事件流 | 用户两次输入的密码不同  系统提示两次输入的密码不相同  用户重新填写并提交。 |

-业务流程

无

-界面设计

界面参照: <https://wwhkiller.github.io/is_analysis_pages/test6_Ui/student.html> API接口调用

接口1：setPassword

算法描述

无

-参照表

USERS

#### “修改用户信息”用例

-用例规约

|  |  |
| --- | --- |
| 用例名称 | 修改用户信息 |
| 功能 | 修改用户的GitHub用户名称 |
| 参与者 | 学生，老师 |
| 前置条件 | 必须先登录，并且查看用户现有的GitHub用户名 |
| 后置条件 |  |
| 主事件流 | 用户填写GitHub用户名称  用户提交修改信息  系统存储修改后的GitHub用户名称 |
| 备选事件流 | 1a. 如果用户输入的GitHub用户名称为空  1.系统清空用户的GitHub用户名称 |

-业务流程

无

-界面设计

界面参照: <https://wwhkiller.github.io/is_analysis_pages/test6_Ui/student.html> API接口调用

接口1：getUserInfo 接口2：setUserInfo

-算法描述

无

-参照表

USERS

#### “查看用户信息”用例

- 用例规约

|  |  |
| --- | --- |
| 用例名称 | 查看用户信息 |
| 功能 | 查看用户的所有信息 |
| 参与者 | 学生，老师 |
| 前置条件 | 必须先登录 |
| 后置条件 |  |
| 主事件流 |  |
| 备选事件流 |  |

- 业务流程

无

- 界面设计

界面参照: <https://wwhkiller.github.io/is_analysis_pages/test6_Ui/student.html> API接口调用

接口1：getUserInfo

- 算法描述

无

- 参照表

STUDENTS

TEACHERS

USERS

#### “绑定平台”用例

-用例规约

|  |  |
| --- | --- |
| 用例名称 | 绑定平台 |
| 功能 | 绑定代码托管平台 |
| 参与者 | 学生 |
| 前置条件 | 每个学生都必须至少绑定一个用来提交作业 |
| 后置条件 | 绑定平台后，选择一个平台用来提交作业 |
| 主事件流 | 绑定平台  输入平台帐号  提交  系统调用接口进行平台绑定 |
| 备选事件流 |  |

-业务流程（顺序图） 源码

无

-界面设计

界面参照: <https://wwhkiller.github.io/is_analysis_pages/test6_Ui/student.html> API接口调用

-算法描述

无

参照表

STUDENTS

#### “登录”用例

-用例规约

|  |  |
| --- | --- |
| 用例名称 | 登录 |
| 功能 | 登录平台 |
| 参与者 | 访客 |
| 前置条件 |  |
| 后置条件 | 登录成功后，跳转到主页 |
| 主事件流 | 访客输入用户名和密码，选择用户类型  系统判断用户名，密码，用户类正确，允许登录  系统在客户端以Cookie形式存储登录用户信息，保持登录的持久性。 |
| 备选事件流 | 1a. 输入的用户名或者密码为空  提示重新输入  访客重新提交登录信息  2a.系统判断用户名，密码，用户类不正确，不允许登录  提示重新输入  访客重新提交登录信息 |

-业务流程

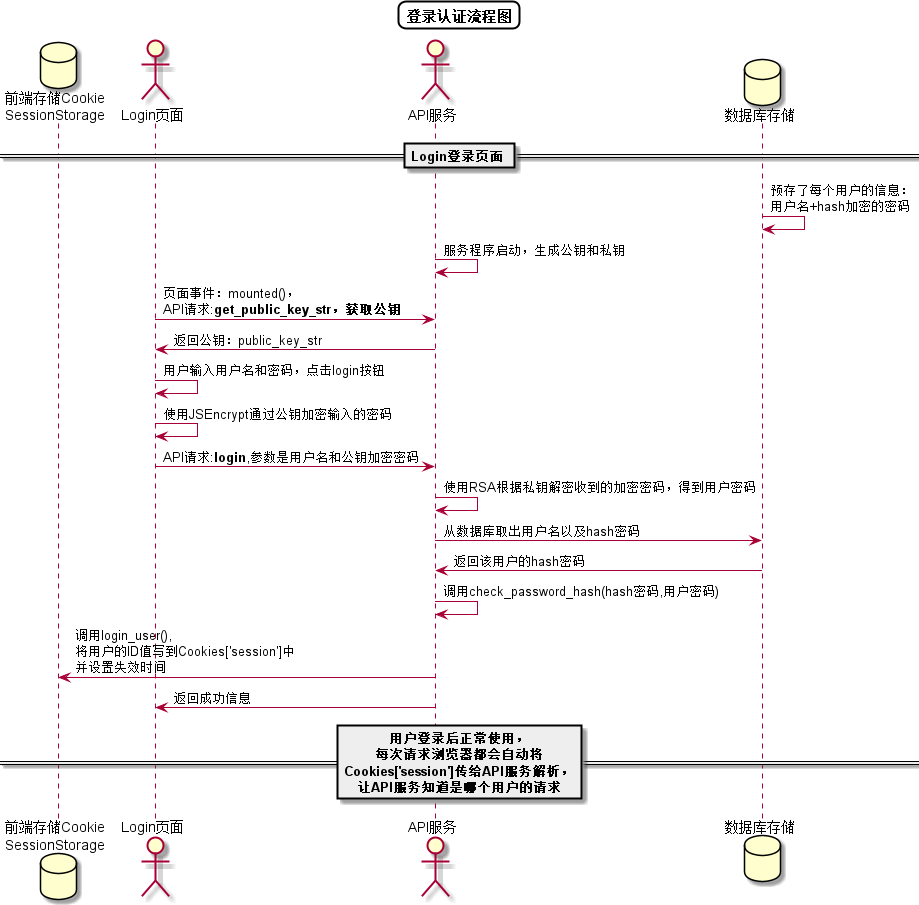
无

-界面设计

-界面参照: <https://wwhkiller.github.io/is_analysis_pages/test6_Ui/login.html> API接口调用

-接口1：login

算法描述 源码



- 参照表

USERS

## 接口

#### 接口：getStudents

权限： 访客：不能看到RESULT\_SUM，WEB\_SUM 老师：可以看到RESULT\_SUM，WEB\_SUM。功能： 返回所有学生的列表。

API请求地址： 接口基本地址/v1/api/getStudents

请求方式 ： GET 请求参数说明: 无返回实例：

{

"status": true, "info": null,

"total": 121,

"data": [

{"WEB\_SUM": "Y,Y,Y,Y,Y,N",

"RESULT\_SUM": "83.75,90,80,80,85,N",

"GITHUB\_USERNAME": "Chinajuedui", "STUDENT\_ID": "201510315203", "CLASS": "软件(本)15-1",

"NAME": "陈松华",

"UPDATE\_DATE": "2018-04-02 13:48:01"},

{

...其他学生

}

]

}

返回参数说明：

#### 接口：getNextPrevStudent

功能： 返回一个学生的上一个或者下一个学生的学号。权限：

老师：只有老师可以调用该API。

API请求地址： 接口基本地址/v1/api/getNextPrevStudent/<is\_next>/<student\_id> 请求方式 ： GET

请求参数说明:

|  |  |
| --- | --- |
| 参数名称 | 说明 |
| is\_next | 是下一个还是上一个，如果为1表示取下一个学生，如果为0表示上一个学生 |
| student\_id | 学生的学号 |

返回实例：

{

"status": true, "info": null,

"student\_id": "201510315203"

}

返回参数说明：

|  |  |
| --- | --- |
| 参数名称 | 说明 |
| status | bool类型，true表示正确的返回，false表示有错误 |
| info | 返回结果说明信息 |
| student\_id | 学号 |

#### 接口：getOneStudentResults

功能： 返回一个学生的所有实验成绩和实验评价。

权限： 学生：只能查看自己的成绩，即接口参数student\_id必须等于登录学生的student\_id 老师：可以查看所有学生的成绩。

API请求地址： 接口基本地址/v1/api/getOneStudentResults/<student\_id> 请求方式 ： GET

请求参数说明:

|  |  |
| --- | --- |
| 参数名称 | 说明 |
| student\_id | 学生的学号 |

返回实例：

{

"status": true, "info": null,

"student\_id": "201510315203",

"github\_username": "chinajuedui", "class": "软件(本)15-1",

"name": "陈松华",

"total": 6, "avgresult":90.5, "data": [

{

"test\_id":1, "web\_exists": true, "result": 91, "memo":"本实验做得好",

"update\_date": "2018-04-02 13:48:01"

},

{

...其他实验

}

]

}

返回参数说明：

#### 接口：getUserInfo

功能： 查看用户的所有信息。

权限： 学生/老师：查看自己的信息，必须先登录，不能查看其他用户的信息。

API请求地址： 接口基本地址/v1/api/getUserInfo/<user\_id> 请求方式 ： GET

请求参数说明:

|  |  |
| --- | --- |
| **参数名称** | **说明** |
| user\_id | 用户的ID号。对应表USERS.USER\_ID的值 |

返回实例：

{

"status": true, "info": null, "ID":"2101201281",

"name":" 张 三 ", "class\_dept":"软件工程1班" "github\_username":"ABCD", "type":"学生"

}

返回参数说明：

|  |  |
| --- | --- |
| **参数名称** | **说明** |
| status | bool类型，true表示正确的返回，false表示有错误 |
| info | 返回结果说明信息 |
| ID | 学号或者工号 |
| name | 用户的真实姓名 |
| class\_dept | 班级或者部门名称 |
| github\_username | gitHub用户名 |
| type | 用户类型：老师或者学生 |

#### 接口：login

功能： 登录到平台。权限： 访客。

API请求地址： 接口基本地址/v1/api/login

请求方式 ： POST

请求实例：

{

"id":"21048329823",

"password":"ABCDE",

"type":"学生"

}

请求参数说明:

|  |  |
| --- | --- |
| 参数名称 | 说明 |
| id | 学生学号或者老师的工号，由type决定。 |
| password | 用户的密码，不是原文，是加密后的字符串 |
| type | 用户类型，学生或者老师 |

返回实例：

{

"status": true, "info": null,

}

返回参数说明：

|  |  |
| --- | --- |
| 参数名称 | 说明 |
| status | bool类型，true表示正确的返回，false表示有错误 |
| info | 返回结果说明信息 |

#### 接口：setOneStudentResults

功能： 设置一个学生的部分实验成绩和评语。

输入参数result不为空，自动设置update\_date为当前日期，表示同时设置批改日期。 输入参数result为空，自动设置update\_date为空，表示未批改。

权限： 老师：可以查看所有学生的成绩。

API请求地址： 接口基本地址/v1/api/etOneStudentResults 请求方式 ： POST

请求实例：

{ "student\_id": "201510315203", "total": 6, "data": [ { "test\_id":1, "result": 91, "memo":"本实验做得好", }, { ...

其他实验 } ] }

请求参数说明:

|  |  |
| --- | --- |
| 参数名称 | 说明 |
| student\_id | 学号 |
| total | 本次批改的所有实验的总数，小于等于全部实验的总数 |
| data | 实验的成绩和评语 |
| test\_id | 实验编号 |
| result | 本实验成绩，可以为空，为空表示没有批改。 |
| memo | 本实验老师的评价，可以为空 |

返回实例：

{

"status": true, "info": null

}

返回参数说明：

|  |  |
| --- | --- |
| 参数名称 | 说明 |
| status | bool类型，true表示正确的返回，false表示有错误 |
| info | 返回结果说明信息 |

#### 接口：setPassword

功能： 修改（设置）用户的密码。

权限： 学生/老师：修改自己的密码，必须先登录。API请求地址： 接口基本地址/v1/api/setPassword 请求方式 ： POST

请求实例：

{

"user\_id":1234, "password":AF#W@#AAAASDF

}

请求参数说明:

|  |  |
| --- | --- |
| 参数名称 | 说明 |
| user\_id | 用户的ID号。对应表USERS.USER\_ID的值 |
| password | 用户的密码。不能为空，必须经过加密码,不能是密码的原文。 |

返回实例：

{

"status": true, "info": null,

}

返回参数说明：

|  |  |
| --- | --- |
| 参数名称 | 说明 |
| status | bool类型，true表示正确的返回，false表示有错误 |
| info | 返回结果说明信息 |

#### 接口：setUserInf

功能： 修改用户的GitHub用户名。

权限： 学生/老师：修改自己的密码，必须先登录。API请求地址： 接口基本地址/v1/api/setUserInfo 请求方式 ： POST

请求实例：

{

"id":"21048329823",

"github\_username":"ABCDE",

}

请求参数说明:

|  |  |
| --- | --- |
| **参数名称** | **说明** |
| user\_id | 用户的ID号。对应表USERS.USER\_ID的值 |
| github\_username | 用户GitHub用户名。 |

返回实例：

{

"status": true, "info": null

}

返回参数说明：

|  |  |
| --- | --- |
| **参数名称** | **说明** |
| status | bool类型，true表示正确的返回，false表示有错误 |
| info | 返回结果说明信息 |